

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**

**BAKALÁŘSKÁ PRÁCE**

2011

Petr Kriegisch

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Absolvování individuální odborné praxe**  
**Individual Professional Practice in the**  
**Company**

2011

Petr Kriegisch

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání bakalářské práce

Student:

**Petr Kriegisch**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**Absolvování individuální odborné praxe  
Individual Professional Practice in the Company**

Zásady pro vypracování:

- 1 Student vykoná individuální praxi ve firmě: Tieto Czech s.r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty

Vedoucí bakalářské práce: **Mgr. Marek Menšík, Ph.D.**

Konzultant bakalářské práce: Ing. Ondřej Kvasnovský

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

### **Poděkování**

Na tomto místě bych rád poděkoval všem zaměstnancům firmy Tieto , zvláště pak svému nadřízenému Ondřeji Kvasnovskému, který mě začlenil do týmu a po dobu trvání odborné praxe vždy s ochotou poradil.

### **Prohlášení**

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Jeseníku dne 26. března 2011

.....

## **Abstrakt**

Bakalářská práce se zabývá popisem, jak probíhala odborná praxe ve firmě Tieto. Pracoval jsem na pozici Java developer a po téměř celou dobu této praxe jsem pracoval na aplikaci Quiz, která má sloužit pro testování uchazečů o práci ve firmě Tieto a taktéž pro srovnání již stávajících zaměstnanců a jejich vědomostí z různých oddělení. Aplikace by měla do budoucna sloužit také pro Tieto IT Academy pro výběr vhodných uchazečů. V téhle práci se zabývám popisem některých mých implementací a následném testování či ladění databáze.

## **Klíčová slova**

Tieto, Quiz, Hibernace, Vaadin, MySQL

## **Abstract**

Bachelor thesis describes how the training is conducted in a company these. I worked on Java developer position and almost the entire duration of this practice, I worked on the application, Quiz, which is used for testing of job applicants in the company of these and also a comparison of existing staff and their knowledge of various departments. The application should also serve the future of IT for The Academy for the selection of suitable candidates. In this work, sion countries deal with the description of some implementation and subsequent testing and tuning the database.

## **Keywords**

Tieto, Quiz, Hibernace, Vaadin, MySQL

## **Seznam použitých symbolů a zkratek**

Atlas – jedno ze sídel firmy Tieto v Ostravě na ulici Výstavní 292/13

Hibernate – Java framework pro práci s databází

HR – oddělení firmy Tieto pro práci se zaměstnanci a novými uchazeči (Human resources)

Java – programovací jazyk

Maven repository – úložiště pro volně dostupné Java knihovny

Ldap – stromová struktura záznamů

TestSet – předpis pro test, který se spustí v aplikaci quiz

Layout – úložiště, ze kterých se skládá stránka ve frameworku vaadin pro jednotlivé komponenty

User – role uživatele quiz, s právy pouze spuštění testu

Vaadin – webový framework pro programovací jazyk Java

Jira – nástroj pro zaznamenávání požadavků k aplikaci

TZD – předmět Teorie zpracování dat

DAIS – předmět Databázové a informační systémy

CSS – kaskádové styly pro úpravu vzhledu webových stránek

JAT – předmět Java technologie

SCRUM – metodika agilního programování

## Obsah

|  |    |
|--|----|
| 1. Úvod .....  | 1  |
| 2. Firma Tieto Czech .....   | 2  |
| 2.1 Odborné zařazení firmy .....   | 2  |
| 2.2 Pracovní zařazení ve firmě .....   | 2  |
| 3. Aplikace Quiz .....   | 3  |
| 3.1 O aplikaci .....   | 3  |
| 4. Úkoly zadané v průběhu odborné praxe s vyjádřením jejich časové náročnosti .....                              | 7  |
| 4.1 Seznámení s frameworky a potřebnými nástroji .....   | 7  |
| 4.2 Obsluha databáze .....   | 7  |
| 4.3 Návrhy na vylepšení .....  | 7  |
| 4.4 Samotná implementace .....   | 7  |
| 4.5 Testování aplikace .....   | 8  |
| 4.6 Návrh nové aplikace .....  | 8  |
| 5. Postup řešení jednotlivých úkolů .....  | 8  |
| 5.1 Seznámení s frameworky a potřebnými nástroji .....   | 8  |
| 5.2 Obsluha databáze .....   | 9  |
| 5.3 Návrhy na vylepšení .....  | 9  |
| 5.4 Samotná implementace .....   | 9  |
| 5.5 Testování aplikace .....   | 14 |
| 5.6 Návrh nové aplikace .....  | 14 |
| 6. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné v průběhu odborné praxe ..... | 15 |
| 7. Znalosti či dovednosti scházející v průběhu odborné praxe .....   | 16 |
| 8. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení .....                                     | 17 |
| 9. Seznam obrázků .....  | 18 |
| 10. Literatura .....   | 19 |



# 1. Úvod

V rámci bakalářského studia jsem si zvolil možnost absolvovat odbornou praxi v jedné z firem, které nabízela škola jako alternativu za klasickou bakalářskou práci s volitelným tématem. Vybral jsem si firmu Tieto, odvětví vývoje Java aplikací, zaslal životopis a druhý den jsem byl pozván k výběrovému řízení, které se konalo 26. července. Na základě pohovoru, který se týkal odborných znalostí a komunikativní schopnosti v anglickém jazyce, který je primární v rámci firmy Tieto jsem byl přijat na místo Java developer. Den nástupu byl stanoven na 1. října. Na začátku jsme absolvovali dvoudenní školení, kde nám byla představena firma Tieto, byli jsme poučeni o vnitřním chodu a bezpečnosti práce.

Byl jsem přiřazen na budovu Atlas, která je jedna ze tří sídel firmy v Ostravě. Po přidělení potřebných přístupových oprávnění jak do systému, tak do budovy nám byla představena aplikace Quiz, na které jsem pracoval po celou dobu trvání praxe.

V první části mé práce se zmiňuji o firmě Tieto, čím se zabývá a jaká bylo mé pracovní zařazení. Následuje stručný popis aplikace, kterou jsem společně s dalšími studenty vyvíjel a použité frameworky, se kterými jsem po celou dobu pracoval.

Popisuji zde, jakými komplexnějšími úlohami jsem se zabýval, a následuje podrobnější popis, co jednotlivé úkoly obnášely.

Zabývám se také tím, jaké znalosti jsem načerpal v rámci studia na VŠB – TU Ostrava a následně je uplatnil v průběhu praxe. Nechybí také druhý úhel pohledu, kde se zabývám naopak tím, co mi ze studia scházelo a v čem mohl spočívat problém mé nevzdělanosti.

Na závěr hodnotím praxi jako celek, včetně výsledků, kterých jsem docílil a zda bych doporučil ostatním studentům možnost volby odborné praxe na místo klasické bakalářské práce.

## **2. Firma Tieto Czech**

### **2.1 Odborné zařazení firmy**

Firma Tieto Czech je nadnárodní korporace, která se zaměřuje na komplexní řešení v oblasti IT služeb. V současnosti zaměstnává firma tisíce IT specialistů, kteří se podílí na vývoji produktů šitých na míru širokému okruhu zákazníků. Kromě poskytování IT služeb se tato společnost orientuje také na výzkum v oblasti podnikání, tzn., snaží se zajistit co nejefektivnější způsoby rozvoje a vývoje produktů. Další ze služeb poskytované firmou Tieto je outsourcing a v neposlední řadě taky cloud computing.

### **2.2 Pracovní zařazení ve firmě**

Když jsem zasílal životopis do firmy Tieto, součástí byl také formulář, ve kterém jsem měl mimo jiné uvést, o jakou pozici se ucházím. Jelikož jsem od prvního ročníku na vysoké škole programoval v jazyce Java a ze všech ostatních jazyků, které jsem měl doposud možnost poznat, mě nejvíce Java zaujala a o jiné pozici jsem ani neuvažoval.

Byl jsem tedy zařazen na pozici Java developer. Jelikož se jedná o velkou firmu tak jsem si byl celkem jistý, že budu vykonávat právě tuhle pozici a žádnou jinou jako tomu může být v menších firmách, kde se dělá to, co je aktuálně potřeba.

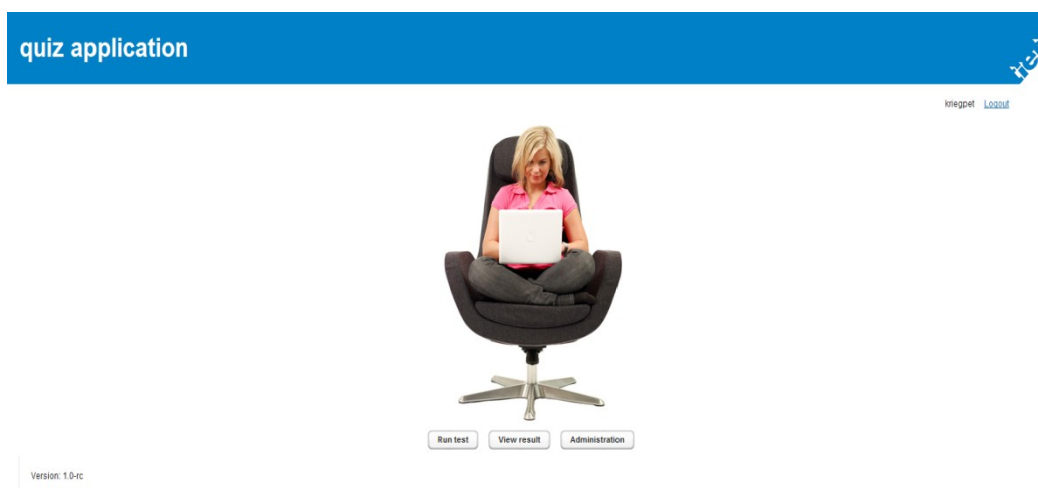
Pozice developera zahrnovala nejen samotnou implementaci ale také návrh, zpracování, vytvoření a ladění databáze a v neposlední řadě důkladné testování a odstraňování chyb v aplikaci.

### 3. Aplikace Quiz

Jedná se o aplikaci, na které jsem po celou dobu trvání odborné praxe pracoval. Byla částečně rozpracována a mým úkolem bylo rozšiřovat stávající funkcionalitu.

#### 3.1 O aplikaci

Aplikace Quiz vznikla za účelem nahrazení stávajícího systému přijímání nových uchazečů o zaměstnání ve firmě Tieto. Původní funkcionalita byla založena na pevně daném počtu otázek pro každého uživatele, který byl testován. Později byl přidán druhý způsob testování a to formou dynamického testu, který se přizpůsobuje podle aktuálního počtu správných odpovědí. Tahle funkce vznikla za účelem rychlejšího vyhodnocování skutečných znalostí a možnosti jakéhosi soutěžení mezi zaměstnanci ve firmě. Jelikož čím větší množství otázek je zodpovězeno správně, tím dále se uživatel dostane.



Obrázek 1: Hlavní okno po přihlášení

#### 3.2 Princip aplikace

##### 3.2.1 Použité technologie

**Apache maven** - je nástroj pro správu, řízení a automatizaci buildů aplikací. Ačkoliv je možné použít tento nástroj pro projekty psané v různých programovacích jazycích, podporován je převážně jazyk Java. (1)

**Vaadin** - je open source webový aplikační framework pro RIA aplikace. Na rozdíl od Javascriptu je vybaven server-side architekturou, což znamená, že většina logiky běží na straně serveru. Ajax technologie se používá na klientské straně. Vaadin může být rozšířen s Google WebToolkitem. (2)

**MySQL** - je databázový systém, vytvořený švédskou firmou MySQL AB, nyní vlastněný společností Sun Microsystems, dceřinou společností Oracle Corporation. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci. (3)

**Hibernate** - je framework napsaný v jazyce Java, který umožňuje tzv. objektově-relační mapování (ORM). Usnadňuje řešení otázky zachování dat z objektů i po ukončení běhu aplikace. Provádí podobné věci jako např. JPA – Java Persistence API. (4)

**GlassFish** - je aplikační server vyvinutý společností Sun Microsystems pro platformu Java EE. GlassFish se řadí mezi open source podléhající licencím GPL a CDDL. GlassFish je referenční implementace, to znamená, že není primárně určen pro provoz aplikací, ale slouží především jako ukázka implementace nových rysů v poslední specifikaci platformy JAVA EE. (5)

### 3.2.2 Sestavení testu

Jednotlivé testy se skládají z několika dílčích částí. Pro vytvoření testu je zapotřebí vytvoření TestSetu, který se skládá z kategorií a levelů. Kategorie jsou vytvářeny k tomu, aby každá vytvořená otázka mohla být přiřazena k jedné kategorii a do jednoho levelu, kterých je celkem 6. Při vytvoření TestSetu je nutné zvolit, zda se bude jednat o staticky či dynamicky vyhodnocovaný test. Pokud se zvolí statický TestSet, je možné vybrat libovolné množství kategorií a libovolné množství levelů. Po zvolení dynamického TestSetu je možné vybrat pouze kategorie. Počet levelů je nastaven automaticky na 6. Jako poslední volbu je potřeba nastavit počet otázek na jednotlivé levely. Při zvolení dynamického TestSetu je počet nastaven na 2 otázky v každém z levelu.

Při vytváření otázky je nutné vyplnit několik povinných údajů. Prvním z nich je, zda se jedná o otázku s jednou správnou odpovědí, více správnými odpověďmi, či odpověď bude formou vlastního textu. Druhý povinný údaj je vlastní text otázky. Dále je potřeba vybrat o kterou kategorii a level se jedná. Lze vždy vybrat jen jednu volbu.

Před samotným uložením otázky je potřeba vložit možné odpovědi a označit, která bude vyhodnocována jako správná.

### 3.2.3 Role uživatelů

Aplikace pracuje se čtyřmi stupni oprávnění. Při spuštění je vždy potřeba vyplnit přihlašovací údaje, podle kterých se vyhodnotí příslušná role.

Pokud není uživatel uveden v databázi aplikace, ale je ověřen vůči smtp serveru Tieta, je mu přidělena role „User“. Takový to uživatel má pouze možnost zvolit si TestSet, který mu přísluší. Po dokončení testu je uživatel automaticky odhlášen.

Druhá z rolí je pro skupinu „HR“, která má stejné možnosti jako „User“, ale navíc může prohlížet veškeré výsledky, které jsou v databázi uloženy.

Třetí role je připravena pro administrátory, kteří mají možnost přidávat do databáze nové otázky a vytvářet tím TestSety pro testování uživatelů.

Poslední role je tzv. „Super admin“, který má možnost kompletní správy celé aplikace. Práva jsou rozšířeny o přidávání nových uživatelů a možnost spravovat veškeré data, které jsou v databázi uloženy.

### 3.2.4 Spuštění testu

Každý uživatel, který se do aplikace přihlásí má možnost spustit jeden z připravených TestSetů. Všechny vytvořené TestSety mají pevně stanoveno, zda se jedná o statický či dynamický TestSet. Takže uživatel ví, jak bude v průběhu testu vyhodnocován.

Pokud se jedná o statickou volbu, u každé otázky je vidno, kolik otázek ještě zbývá do konce.

Když se jedná o dynamickou formu, je potřeba ze dvou otázek, které jsou v každém levelu odpovědět alespoň na jednu správně, aby mohl uživatel pokračovat dále. Pokud jsou obě zodpovězeny špatně, je ukončena aktuální kategorie a uživatel přechází k další kategorii, ze které je TestSet složen. Jestliže už byla dokončena poslední kategorie, test je ukončen.

The screenshot shows a web application interface for a quiz. At the top, there is a blue header bar with the text "quiz application" on the left and a small "test" logo on the right. Below the header, the main content area is white. It starts with "Question 1 / 18". Underneath, it says "Category: Java Level 1" and "single-choice: You can choose only one correct answer." The question text is "A class cannot be declared:". Below this, there are three radio button options: "Static", "Private", and "Default". At the bottom of the question area, there is a text box with the placeholder "You can write your comment (maximum is 255 characters)" and a "Next" button.

Obrázek 2: Otázka statického testu

### 3.2.5 Zobrazování výsledků

K prohlédnutí dokončených testů má oprávnění každý uživatel, který není v roli „User“. Dále pak skupina uživatelů v roli „HR“ a „Super admin“ má právo prohlížet veškeré testy, které jsou

v databázi uloženy. Uživatelé v roli „Admin“ mají oprávnění prohlížet pouze testy, které se týkají TestSetů jimi vytvořenými.

Pro vyhledávání slouží čtyři kritéria. Prvním z nich je počáteční datum. Uživatel si zadá, od jakého období chce sledovat výsledky a odpovídající testy se mu zobrazí. Druhou možností je zadat koncové datum a zobrazí se všechny odpovídající výsledky. Další kritérium je filtrování výsledků podle uživatele, který test dokončil. Poslední možnost jak vyhledat hotový test je vyplněním údaje o uživateli, který na test dohlížel.

Další volbou je možnost exportovat hotové výsledky do formátu PDF. Což je ideální pro možnosti tisku či snadného archivování výsledků.

Každý hotový test je také možné odeslat emailem na adresu uživatele, který byl přihlášen. Jako odesílatel je uveden uživatel, který je aktuálně přihlášen.

Jednotlivé testy jdou zde také mazat.

quiz application

kriegpetLogout

Results of completed tests







Start date: 2011-03-03

End date:

Trainee:

Examiner:

search

| ID | MENU  | TRAINEE  | EXAMINER | TEST SET | DATE       | RESULT | SUCCESS RATE (%) | TIME SPENT |
|----|---|----------|----------|----------|------------|--------|------------------|------------|
| 2  |    | kriegpet | kriegpet | static1  | 2011-05-03 | 0/18   | 0.0              | 00:03:03   |
| 1  |    | kriegpet | kriegpet | static1  | 2011-04-22 | 0/18   | 0.0              | 00:00:17   |

Back

Obrázek 3: Zobrazení výsledků

## **4. Úkoly zadané v průběhu odborné praxe s vyjádřením jejich časové náročnosti**

Aplikace quiz již byla rozpracována. Tři studenti jsme tedy převzali tento projekt, který měl sloužit pro interní potřeby a my ho měli rozšiřovat o další funkcionality. Počátkem nového roku k nám byli přiděleni další dva studenti, se kterými jsme pracovali na celkovém dokončení aplikace.

Před námi vyvíjeli projekt čerství absolventi bakalářského studia, takže chyběla většina komentářů či dodržování jakési konvence čistého programování. Taktéž databáze nebyla navrhnutá zrovna nejlépe, ale našim úkolem bylo během měsíce či dvou dodělat základní funkcionalitu aby mohl být projekt nasazen.

### **4.1 Seznámení s frameworky a potřebnými nástroji**

Firma Tieto pracuje s různými frameworky a nástroji. Proto jsem se po dobu celé praxe vzdělával a poznával věci, které jsou v dnešním IT světě aktuální.

Mezi první mé úkoly patřilo nainstalovat si všechny potřebný software, abych mohl začít pracovat. První dva dny jsem tedy instaloval, nastavoval a seznamoval se s nástroji, se kterými se budu po celou dobu setkávat a efektivně používat.

### **4.2 Obsluha databáze**

Quiz běží na databázi MySQL a po celou dobu bylo potřeba udržovat databázi v chodu a použitelnou pro všechny uživatele. Mým úkolem bylo téměř každý den na databázi provádět jednotlivé úpravy a optimalizace.

### **4.3 Návrhy na vylepšení**

Úkolem developera není pouze programovat dle zadané specifikace, ale také přispět k vylepšení, či usnadnění samotné obsluhy aplikace a tím zefektivnit vývoj projektu. Prostor pro případné návrhy a vylepšení byl každý týden na společném meetingu, kde se diskutovalo o případných změnách.

### **4.4 Samotná implementace**

Každý týden jsme po společném meetingu měli přesně stanoveny, co má kdo dělat a jednotlivé úlohy byly zadány do nástroje, který slouží pro účely Tietu, kde jsou uloženy projekty a k nim jednotlivé požadavky.

## 4.5 Testování aplikace

I když ke každému projektu jsou přiděleni testéři, mým úkolem bylo vždy po každé vytvořené funkcionalitě náležitě otestovat, zda vše pracuje jak má a neovlivnilo se mým zásahem nic jiného. Samotné testování a následné opravování chyb mnohdy zabíralo mnohem více času než samotná implementace.

## 4.6 Návrh nové aplikace

Když jsme dokončili první fázi vývoje aplikace Quiz zdálo se, že budeme přiděleni k novému projektu, který měl také sloužit pro interní účely. Požadavek vzniknul na začátku měsíce února a začali jsme na tom pracovat všichni studenti. Naším úkolem nebylo v tak krátkém časovém úseku dokončit kompletní projekt, ale navrhnout vzhled a vše, co bylo potřebné pro začátek programování.

## 5. Postup řešení jednotlivých úkolů

### 5.1 Seznámení s frameworky a potřebnými nástroji

Jak jsem již uvedl, aplikace byla rozpracovaná a trvalo mi několik dní, než jsem proniknul do logiky celého systému a zorientoval se v jednotlivých třídách. Vše umocňoval použitý framework Vaadin se kterým jsem se do té doby nesetkal. I když se to na první pohled zdálo jednoduché, tak bylo potřeba nastudovat pár tutoriálů a doptávat se ostatních spolupracovníků. Po prvním měsíci si troufám říct, že jsem byl schopný upravit či dodělat cokoliv, co se týkalo samotného Vaadinu a to díky dobře zpracované podpoře, kde na webových stránkách jsou uvedeny všechny komponenty a jejich ukázkové kódy.

Další pro mě do té doby neznámý framework byl Hibernate pro práci s databází. Mapování databáze bylo provedeno přehledně, takže nastudovat DAO vrstvu trvalo jeden den. Jednotlivé zásahy do mapování tedy nebylo problém dodělat.

Velice důležité pro spolupráci na projektu bylo naučit se správně používat sdílení projektu mezi spolupracovníky za pomoci SVN. Díky tomu, že jsem pracoval v programu NetBeans tak stačilo doinstalovat plugin pro sdílení projektu a pochopit princip commitování a updatování. Když jsem zjistil jak silný nástroj tohle je, začal jsem používat SVN i pro školní projekty už jen kvůli verzování a nemusel jsem tak mít strach, že o něco přijdu.

Dalším nepostradatelným nástrojem je tzv. JIRA, kde se zadávají jednotlivé úkoly a loguje se tam také práce, kterou jednotliví programátoři udělali na daném požadavku. Při vytvoření např. nové funkcionality se zde запиše požadavek a člověk, který je aktuálně volný, si tuhle úlohu přiřadí a napíše si, kolik času odhaduje, že na tom stráví. Po dokončení запиše čas, který trval pro vytvoření funkcionality a přidá komentář o tom, co vytvořil. Tento nástroj funguje také pro zpětnou vazbu od testerů, kteří zde logují chybné funkce a nové požadavky.



Po celou dobu trvání odborné praxe jsem měl k dispozici veliké množství literatury v podobě knih zabývajících se nejrozličnějšími technikami programování a frameworky. Často jsem nahlížel do knížky, která radila jak efektivně programovat v jazyce Java. (6)

## 5.2 Obsluha databáze

Vzhledem k tomu, že ve škole jsme se s databázemi potýkali téměř každý semestr, tak nebyl problém spravovat a udržovat naši databázi.

Při nástupu byla u Quizu vytvořena jedna databáze, která běžela současně s aplikací na serveru v jiné kanceláři u našeho nadřízeného. Ze začátku jsme v tom neviděli žádný problém a všichni pracovali s touhle jednou instancí. Ovšem časem se stávalo, že špatným zásahem do databáze byla celá „shozena“ a ne vždy byla možnost vše hned obnovit.

Zašlo to až tak daleko, že databáze byla kompletně zničena a my tak přišli o všechny data, které v ní byly uloženy. I když se jednalo o testovací data tak bylo dosti pracné nahrát vše zpět. To nás donutilo vytvořit si lokální databáze a každý tak pracoval se svou vlastní. Také jsme si vytvořili spouštěcí skript, který při jakémkoliv problému uvedl databázi do původního stavu, a tím jsme zamezili spoustě problémů.

Jak se aplikace rozvíjela tak bylo neustále nutné upravovat jednotlivé tabulky nebo vytvářet nové. Postupem času jsme se snažili eliminovat všechny nešikovné návrhy databáze tak aby byla její propustnost co největší, jelikož většina funkcionality aplikace spočívá v komunikaci s databází.

## 5.3 Návrhy na vylepšení

Již od nástupu na praxi byly pravidelně každý pátek organizovány meetingy, na kterých měl hlavní slovo náš nadřízený. Projížděl zdrojové kódy a hodnotil, jak je vytvořený kód napsaný.

Vzhledem k tomu, že v rámci školy nebyl kladen na strukturu kódu až takový důraz, tak ze začátku bylo chyb hodně. Snažil jsem se z toho ponaučit a stejných chyb se pro příště vyvarovat.

Když jsme každý týden hodnotili, co se zvládlo udělat a co ještě schází naimplementovat, byl prostor také pro návrhy, jak celou aplikaci uživatelsky zpříjemnit a hlavně zjednodušit. Nejednou se nám stalo, že vytvořený vzhled byl kompletně předělán, jelikož byl pro potencionální uživatele až příliš složitý. Proto byl důležitý pohled na vzhled člověka, který se s funkcionalitou setkal poprvé. Během týdne když jsem přišel na způsob jak cokoliv vylepšit, domluvil jsem si krátkou schůzku s naším nadřízeným a případné změny jsme prodiskutovali.

## 5.4 Samotná implementace

### 5.4.1 Export do PDF

Jak jsem již uvedl výše, po nastoupení na praxi jsme převzali již rozdělaný projekt. Mým prvním úkolem bylo vytvořit funkci pro exportování hotových testů do formátu PDF z důvodu snadnější archivace a přehlednějšího procházení jednotlivých výsledků. Kolega pracoval také na exportu do

wordu, který funguje na podobném principu, ale z důvodu editovatelnosti exportovaného dokumentu byla takhle funkce nakonec zakázána. Můj první odhad činil asi 8 hodin, během kterých jsem myslel, že budu schopen daný problém vyřešit.

Na počátku se to zdálo jako jednoduchá utilita, ale postupně jsem narážel na spoustu problémů. Prvním z nich bylo použít nějakou knihovnu, která není vázána žádnou licencí a nachází se na maven repository. Téměř všechny vzorové příklady pracovaly s knihovnami, které nebyly free a tudíž pro mě byly naprosto nepoužitelné. Až knihovna „iText“, která byla volně dostupná a umožňovala práci s PDF, byla nakonec použita.

PDF dokument se skládá z jednotlivých odstavců, který jsou vyplněny řetězci a nadefinovány různými styly písma. Jednotlivé údaje se získávají z instance hotového testu, který má být exportován.

Druhý problém vznikl při snaze poslat klientskému prohlížeči PDF dokument, který byl sestaven. Jelikož aplikace běží na serveru, tak jakékoliv ukládání souborů nebo jiné uchovávání obsahu na serverové straně je naprosto vyloučeno. Bylo tedy potřeba převést dokument na pole bytů, které za pomoci streamu předat klientskému prohlížeči.

Výsledek otestované funkcionality byl po 25 hodinách hotov. Což oproti původnímu odhadu byl 3x větší nárůst. Postupem času jsem se naučil mnohem přesněji odhadovat náročnosti úkolu.

### 5.4.2 Dynamický algoritmus

Původní algoritmus vyhodnocování testu byl založen na pevně daném počtu otázek u každého testu. Uživatel přesně věděl, na kolikáté otázce se nachází a kolik otázek mu ještě zbývá do konce.

Na jednom z pravidelných meetingů, které probíhaly každý pátek ohledně konzultace pokroku na aplikaci a vytvoření nových požadavků, byl našim managerem vznesen návrh, jak testovat uživatele úplně odlišným způsobem tak, aby se podle znalostí přizpůsoboval test za „chodu“. Měli jsme týden na to, abychom navrhly, jak něco takového zrealizovat. Odhad činil 16 hodin.

Vznikl tedy úplně nový algoritmus, který po každé zodpovězené otázce zjišťoval, zda alespoň jedna ze dvou otázek daného levelu je správně. Pokud ano, mohl se uživatel posunout a level výše. Pokud ne, test byl ukončen.

Implementace spočívala ve správném nastavování jednotlivých stavů a počítání modula, které bylo nastaveno podle počtu otázek v levelu. Pokud tedy nastala situace, kdy alespoň jedna z otázek byla správně, aplikoval se stav „State\_Question“. Než bylo tak učiněno, vždy se ověřovalo, zda se v testu ještě nějaká další otázka nachází. Pokud ne, aplikoval se stav „State\_Last“ a test se tak mohl ukončit.

Posledním krokem bylo upravit jednotlivé stavy v závislosti na typu testu. U statického testu se mohl uživatel vracet k již zodpovězeným otázkám, což u dynamického testu bylo vyloučeno.

Pracovali jsme na tom ze začátku s kolegou společně, ale během pár minut jsme zjistili, že takovým způsobem se k výsledku určitě nedostaneme. Každý si tedy vzal dílčí úkol a při společném sdílení jednotlivých výsledků jsme se dopracovali k prvnímu algoritmu po 16 hodinách. Několik hodin trvalo, než jsme odladili všechny chyby a mohli tak dát funkčnost testerům na důkladnější otestování.

### 5.4.3 Dokumentace k aplikaci

Za běhu času se objevovali různé nápady na obohacení funkcionality. Bylo to dáno tím, že na začátku nebyla vytvořena žádná dokumentace k tomu, jak by aplikace měla vlastně vypadat a co přesně by měla dělat. Bylo to zapříčiněno hlavně tím, že aplikace má sloužit pro interní potřeby a tudíž přesné požadavky neměl kdo určit. Původně to měla být jednoduchá aplikace, od které se očekávalo, že do měsíce bude nasazena a používána. Vzhledem k tomu, že až v průběhu implementace vznikaly složitější problémy, stala se aplikace nakonec propracovanější, než se vůbec čekalo.

Mým dalším úkolem tedy bylo vytvořit přesný manuál popisující jednotlivé funkce a možnosti pro všechny uživatele, kteří budou výsledný projekt používat. Tieto pro tyto účely má vytvořeno vlastní prostředí, které pracuje na podobném principu jako např. Microsoft Word. Lze tam jednoduše stylovat, nahrávat obrázky, vytvářet odkazy mezi dokumenty a sdílet jednotlivé verze vytvořeného dokumentu. Velkou výhodou je jakési verzování. Jelikož na jednom dokumentu může pracovat více lidí najednou, jednotlivé úpravy jsou zaznamenávány a uživatel se tak může podívat, co se oproti jeho verzi změnilo a kdo změnu provedl.

Vytvořil jsem tedy podle jednotlivých rolí v aplikaci podrobnou dokumentaci včetně obrázků a tím i přesnější vodítko pro testery, kteří byli na projekt nasazeni. Vytvoření celého dokumentu trvalo asi 20 hodin, jelikož bylo nutné psát vše anglicky.

### 5.4.4 Odesílání emailů

Dalším požadavkem, který vzniknul až v průběhu, bylo odesílání hotových testů uživateli, který test dokončil. Všechny potřebné informace se v databázi nacházely, takže zbývalo už jen začít programovat. Odhadoval jsem 8 hodin

Použili jsme JavaMail API, které nám poskytlo veškeré potřebné funkce. Pomocí LDAPu na základě loginu je zjištěna emailová adresa, na kterou má být email odeslán a test se vybere na základě id z databáze. Jako formát přílohy, která se odesílá, je nastaveno PDF. Po stisku ikonky demonstrující email je zaslána zpráva s přílohou.

Největší úskalí, na které jsme narazili, bylo najít správné nastavení smtp serveru pro odeslání emailu v síti Tieto a také nastavit správné formátování přílohy. Jelikož před samotným odesláním emailu bylo potřeba nejdříve dokument vyexportovat, nastal problém s připojení přílohy, která nemohla být formou souboru, jelikož na serveru ukládat nelze. Museli jsme to tedy opět vyřešit držením dokumentu ve streamu, ve kterém bylo pole bytů. Po úspěšném odeslání emailu se uživateli zobrazí upozornění, že vše proběhlo v pořádku.

### 5.4.5 Odhlašování z aplikace

Vzhledem k tomu, že původní koncept aplikace byl velice jednoduchý, tak se ani s odhlašováním nepočítalo. Uživatel zůstal přihlášen do té doby, než mu vypršela session. Což bylo samozřejmě velice nešikovné a takhle funkcionality se musela doprogramovat. Řešení bylo na nás.

Jelikož se aplikace skládá ze 4 různých náhledů, nebyl velký problém odhlašování dodělat. Vzhledem k tomu, že Vaadin je komponentní framework, tak umístění jednotlivých komponent je velice pracné. Jednotlivé stránky jsou poskládány z tzv. layoutů, které se navzájem skládají do sebe a umísťují se do nich jednotlivé komponenty. Ovšem při přidání jakéhokoliv objektu do layoutu může způsobit úplné rozhození ostatních komponent. V čemž také spočíval vždy největší problém při zásahu do vzhledu. Samotné odhlášení bylo nainplementováno opačnou logikou než přihlášení, kde při startu aplikace je hlavní okno překryto logovacím oknem do té doby, než dojde ke správnému ověření uživatele. Po té dojde k odkrytí hlavní stránky aplikace. Při odhlášení je aktuální stránka překryta opět logovacím oknem a uživatel tak nemá možnost provádět žádné úkony. Kolega navrhnul umístění odhlašovací ikony, vedle které byl vypsán aktuálně přihlášený uživatel, a já dotvořil funkčnost. Veliký problém my dělalo znovu načtení aplikace, jelikož se jednotlivé komponenty musely znovu inicializovat a k tomu nebyla aplikace přizpůsobena. Nakonec vše vyřešilo až odstranění všech prvků z hlavního okna a znovu načtení aplikace. Po zhruba 6 hodinách bylo odhlašování funkční a připraveno k testování.

#### **5.4.6 Editace otázek**

Editování otázek uložených v databázi nebylo možné, a proto vznikl požadavek na úpravu dosavadního administračního modulu, kde by se daly jednotlivé otázky upravovat popřípadě mazat. Muselo se tedy upravit stávající databázi a udržovat si tak informaci o tom, zda byla již otázka použita v nějakém testu či nikoli. Jednotlivým otázkám bylo možné měnit pouze jejich použitelnost do dalších testů. To znamená, že pokud byla otázka nastavena jako invisible, tak bylo zajištěno, aby se již otázka nikdy nepoužila. Tuhle ořezanou funkcionalitu bylo potřeba rozšířit o možnost změny všech atributů. Pokud otázka ještě nebyla použita tak je editace dovolena. Tohle omezení na nepoužité otázky je z toho důvodu, aby byly podmínky pro všechny stejné a pokud je již v databázi otázka uložena a nějaký uživatel na ní odpověděl, aby zůstala stejná pro všechny ostatní. Pro nepoužité otázky teda vznikla možnost jejich editace. Měnil lze všechny její atributy. Včetně odpovědí, které se vážou k dané otázce.

#### **5.4.7 Vyhledávání v otázkách**

Na jednom z pravidelných meetingů jsem dostal úkol vytvořit v administračním modulu na záložce otázek vyhledávání, aby se tak dalo jednodušeji filtrovat otázky v databázi. Čas odhadovaný pro implementaci byl odhadnut na 8 hodin.

Jelikož se otázky dělí podle kategorie, ve které se nacházejí a podle levelu (obtížnosti), vytvořil jsem možnost vybírat jednotlivé možnosti. Vyhledávat lze jednotlivě nebo průnikem obou možností. Označovat jde více než jedna volba v jednotlivých filtrech, akorát problém byl v komponentách Vaadinu, které neposkytovali potřebné funkce pro můj návrh. Musel jsem tedy přidat tlačítko, které po stisku ověřilo vybrané volby a podle toho vyhledalo. Částečně se to také vyřešilo přidáním funkce, která reagovala na stisk klávesy „Enter“ a tím se začalo vyhledávat.

Po dokončení klasického vyhledávání vznikl ještě požadavek na fulltextové vyhledávání v otázkách. Přidalo se tedy textové pole, do kterého se požadovaný řetězec zadal a mohlo se to kombinovat i

s ostatními filtry. Opět na stisk klávesy „Enter“ bylo vyhledávání spuštěno. Fulltextové vyhledávání funguje na principu procházení textu všech otázek, a pokud zadaný řetězec otázka obsahovala, byla zobrazena. Původně se rozlišovala i velikost písma ale po upozornění od testerů to bylo ošetřeno na nezávislost velikosti vstupního řetězce a textu otázky. Po 15 hodinách vše vyřešeno.

The screenshot shows a search interface with the following elements:

- A text input field labeled "Search:".
- A list box labeled "Choose categories \*" containing: Java, .NET, Xml, Anglicky jazyk, SQL, Oracle.
- A list box labeled "Choose levels \*" containing: Level 1, Level 2, Level 3, Level 4, Level 5, Level 6.
- A button labeled "Find".

Obrázek 4: Vyhledávání v otázkách

#### 5.4.8 Sdílení kategorií

V rámci izolovanosti jednotlivých administrátorů bylo dáno, že každý administrátor viděl pouze to, co si sám vytvořil. Neměl tedy možnost prohlížet či spravovat cizí data. Mělo to své opodstatnění z důvodu bezpečnosti, aby se nemohlo stát, že si nepověřená osoba bude prohlížet, nebo dokonce mazat otázky, ke kterým by neměla mít přístup z důvodu férovosti vůči ostatním. Na druhou stranu by se tedy mohlo stát, že budou vznikat podobné nebo dokonce stejné otázky, což bylo nežádoucí. Vyřešil jsem to tedy sdílením kategorií, jelikož i jednotlivé zobrazování otázek závisí na oprávnění administrátora vůči kategoriím. Stanovil jsem si pro to dobu 4 hodiny. Na začátku bylo potřeba navrhnout vzhled a jednoduchou obsluhu tak, aby to bylo co nejsnadnější. Vaadin podporuje velké množství nejrozličnějších komponent, takže nebyl problém vybrat si tu nejvhodnější.

Při vytvoření nové kategorie si tedy může administrátor zvolit, kteří další administrátoři mají možnost prohlížet či měnit danou kategorii. Pokud usoudí, že nějakého o tuhle možnost chce připravit, může ho opět od sdílení odebrat. Samotného tvůrce kategorie nelze odebrat, protože v databázi je jeho id uloženo pevně. Implementace byla po 3 hodinách dokončena.

The screenshot shows a category sharing interface with the following elements:

- A label "Choose admins".
- A list box containing: kriegpet, ribkajak, baraktom, kvasnond, kaisesar, heczkmar, vantutom, zitniond, schifjan, sekulbar.
- Two buttons: ">>" and "<<".
- A large empty list box for the selected administrators.
- A button labeled "Save".

Obrázek 5: Sdílení nové kategorie

## 5.5 Testování aplikace

Po převzetí projektu nebyl do té doby nasazen žádný tester na aplikaci, a proto jsme ji převzali i s některými chybami. První dva testeři byli nasazeni asi po dvou měsících našeho programování. První dva týdny po jejich testování jsem nedělal nic jiného, než opravoval chyby v aplikaci. Některé z nich byly spíše kosmetické, ale některé byly závažné a bylo potřeba je co nejdříve odstranit. Po první fázi testování jsme měli funkční verzi, která byla připravena.

Každý z programátorů když vytvořil novou funkci, měl za úkol si to náležitě otestovat. I já tak činil, ale zdaleka ne všechny problémy jsem dokázal objevit. Nalezené problémy od testerů se zaznamenávali do nástroje JIRA, která sloužil právě pro podobné účely. Zapsán byl problém, který byl nalezen, důkladný popis a pokud bylo potřeba tak i obrázek s jasným vyjádřením co je špatně. Já se pak mohl podívat, co je potřeba opravit, přiřadit si tenhle úkol ke své osobě a zaznamenat jakou dobu budu přibližně pro odstranění potřebovat. Po úspěšném vyřešení jsem zapsal svůj komentář s popisem, co jsem udělal za změny a opět poslal zpět testerům na otestování. Pokud našli další chybu, poslali mi to zpět. Tahle se to opakovalo do té doby, než vše fungovalo, jak mělo.

## 5.6 Návrh nové aplikace

Když jsme byli informováni o novém projektu, bylo třeba uspořádat několik sezení s naším nadřízeným, aby nám zadal přesné požadavky k aplikaci. Mělo se jednat o aplikaci, která bude pracovat s databází a bude sloužit jako přesný plán umístění jednotlivých zaměstnanců firmy Tieto v budovách. Vzhledem k tomu, že firma sídlí ve třech budovách ve městě Ostrava, tak při přijmutí např. nového zaměstnance je složité najít volné místo a začlenit do správného týmu. Pro tyto účely sloužil dokument v Excelu, který uchovával všechny zaměstnance a jejich umístění. Bylo to velice nepřehledné a větší přesuny byly časově velice náročné pro člověka, který tohle měl na starost.

Vše jsme si pečlivě zapisovali a krom funkčnosti nebyly kladeny žádné požadavky.

Na nás tedy bylo navrhnut kompletní vzhled jednotlivých stránek a zpracovat to pro uživatele co nejvíce přívětivě. Taktéž volba veškerých frameworků byla na nás a ze strany vedení nebyly kladeny žádné požadavky.

Vzhledem k tomu, že jsme měli již velice dobře nastudovaný Vaadin, zvolili jsme opět tuhle variantu. Databáze byla zvolena jako MySQL a mapování opět hibernate, jelikož se nám velice zalíbil. Vše nám bylo schváleno a my se tedy pustili do tvoření návrhů na vzhled. Každý si vzal na starost jednu záložku a během týdne byla práce dokončena.

Já pracoval na záložce, která měla sloužit k registraci nových zaměstnanců a popřípadě jejich editaci. Návrh spočíval v grafickém rozmístění jednotlivých komponent a jejich barevném sladění. Bylo třeba znát framework, ve kterém se měla aplikace implementovat kvůli tomu, aby nebyly použity některé komponenty, které nebyly podporovány.

Když bylo vše hotové, tak bylo zjištěno našim nadřízeným, že u aplikace quiz bude třeba dodělat některé funkcionality a tudíž byl nový projekt zastaven a na jeho pokračování už bohužel nebyl čas.

## **6. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné v průběhu odborné praxe**

Při nástupu do prvního ročníku jsem měl s programováním základní znalosti. S programovacím jazykem Java jsem se setkal poprvé až v prvním semestru, kde se probíraly úplné základy. Až teprve ve druhém semestru jsme začali brát objektově orientované programování, což byla pro mě úplně nová věc. Na tyto poznatky navazovaly již další předměty, které mě seznámily s jazykem mnohem více, a troufám si říci, že dostatečně na to, abych mohl obstojně začít pracovat ve firmě Tieto na pozici Java developer.

Další důležité znalosti se týkaly databázové vrstvy, kde téměř veškeré aplikace spolupracují s databází. V předmětu TZD jsem se naučil jak databázi navrhnout tak aby fungovala a byla schopná k nasazení k nějakému jednoduchému informačnímu systému. Zřejmě nejdůležitějším předmětem byl DAIS. Zde jsem se naučil jak databázi propojit na nejvyšší úrovni s vlastním informačním systémem použitím vlastního mapování.

Vzhledem k tomu, že aplikace Quiz byla webovou aplikací, tak bylo výhodou, že jsem uměl za pomoci CSS upravovat vzhled. Bez této znalosti by se těžce manipulovalo s jednotlivými styly.

Díky předmětu JAT pro mě webová aplikace nebyla žádnou novinkou. Rozdíl byl pouze v použitém frameworku, ale podstata byla stejná. Do té doby jsem neuměl pracovat se serverem a spouštět v něm aplikace. Což jsem se v tomto předmětu naučil a ve firmě Tieto to bylo velice užitečné.

## **7. Znalosti či dovednosti scházející v průběhu odborné praxe**

Co se týkalo samotného programování a pracování s databází, byl jsem připraven velice dobře. Pochopitelně jsem nemohl znát každý Java framework a tak byl Vaadin pro mě novinkou. Pochopit a naučit se s ním pracovat nebylo až tak náročné.

Mapování na databázi pomocí hibernate si myslím, že klidně mohlo být v osnovách, protože ve větších aplikacích se používá. Teda aspoň co já jsem měl možnost v rámci firmy Tieto poznat. Znal jsem obdobu v podobě JPA, ale vzhledem k tomu, že jsme nezačínaly nový projekt, nebyla možnost uplatnit tuhle znalost.

Co se týká firmy Tieto, tak je zde úředním jazykem angličtina a ta byla trochu problém pro mě. Absolvoval jsem 4 semestry kurzy angličtiny v rámci školy a myslím, že základ jsem měl dobrý díky odborné angličtině, která se vyučovala. Ovšem značné komunikační problémy se vyskytovaly.

V čem ovšem vidím zásadní nedostatek, který mě ihned po nástupu a seznámení s ostatními zaměstnanci zaskočil, byla neznalost JUnit testů. Překvapuje mě, že za dobu studia jsem nebyl v rámci žádného předmětu seznámen s testováním kódu. S tímhle problémem jsem se potýkal po celou dobu praxe a stejně tak mí kolegové, kteří absolvovali praxi zároveň se mnou.



## 8. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Jsem nesmírně rád, že naše fakulta nabídla možnost absolvovat odbornou praxi místo na místo psaní bakalářské práce. Jsem přesvědčen o tom, že tahle možnost by měla být pro všechny studenty což bohužel nelze zajistit. V rámci školy student nemá moc možností získat praktické znalosti. Pouze se soustřeďuje na teoretické znalosti a školní projekty. Díky odborné praxi jsem si konečně mohl osahat, jak programování probíhá v reálném životě a zda bych mohl po dokončení studia nastoupit do jedné z IT firem.

Další důležitou věcí je získaná praxe, která mě posouvá trochu výše vůči čerstvým absolventům. Absolvoval jsem také školení na téma SCRUM což mi ukázalo další pohled na práci v týmu. Než jsem nastoupil k praxi, myslel jsem si, že práce v týmu je jednoduchá a zvládne to každý. Opak je pravdou. Je velice těžké u nového týmu začlenit každého jedince do týmu, aby byl právoplatným členem hned od začátku. Důležitá je komunikativnost a velkorysost všech jedinců. Každý projekt je týmová práce a výkon celého kolektivu záleží na každém členovi. Než jsem to zažil tak jsem si nedokázal představit, jak takové zorganizování práce může probíhat.

Naučil jsem se používat nástroj pro sdílení projektu mezi ostatní spolupracovníky a tím získat představu o vývoji velkých aplikací.

V neposlední řadě bylo přínosné, že veškerá komunikace s vedením probíhala v angličtině, což je pro mě velikou zkušeností.

Když srovnám aplikaci, kterou jsme dostali na začátku s aplikací, která je jako výsledek za naší práci, musím říct, že jako studenti jsme odvedli veliký kus práce. Při rychlé ukázce naší aplikace jinému oddělení ve firmě Tieto byl pozitivní ohlas a zájem o používání aplikace. Což mě velice potěšilo. Pokud bych měl zpětnou volbu mezi odbornou praxí nebo bakalářskou prací, určitě bych znovu zvolil praxi a doporučil to všem ostatním studentům.

## 9. Seznam obrázků

|   |    |
|---|----|
| Obrázek 1: Hlavní okno po přihlášení..... | 3  |
| Obrázek 2: Otázka statického testu .....  | 5  |
| Obrázek 3: Zobrazení výsledků.....        | 6  |
| Obrázek 4: Vyhledávání v otázkách.....    | 13 |
| Obrázek 5: Sdílení nové kategorie .....   | 13 |

## 10. Literatura

1. Apache maven. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, last modified on 19. 3. 2011 [cit. 2011-04-23]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Apache\\_maven](http://cs.wikipedia.org/wiki/Apache_maven)>.
2. Vaadin. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, last modified on 19.4.2011 [cit. 2011-04-23]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Vaadin>>.
3. Mysql. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, last modified on 7. 4. 2011 [cit. 2011-04-23]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Mysql>>.
4. BAUER, Christian; KING, Gavin . *Hibernate in Action : A guide to the concepts and practise of object/relational mapping*. Manning Publications Co., 2005. 400 s. ISBN 1932394-15-X.
5. Glassfish. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, last modified on 23. 1. 201 [cit. 2011-04-23]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Glassfish>>.
6. BLOCH, Joshua . *Java efektivně : 57 zásad softwarového experta*. GRADA Publishing, a.s., 2002. 232 s. ISBN 80-247-0416-1.